

10/649,307

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 3 年    2 月 2 8 日  
Date of Application:

出 願 番 号                      特 願 2 0 0 3 - 0 5 4 2 7 7  
Application Number:

[ST. 10/C] :                      [ J P 2 0 0 3 - 0 5 4 2 7 7 ]

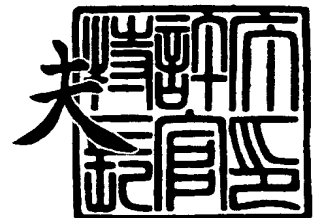
出      願      人                      株式会社日立製作所  
Applicant(s):



2 0 0 3 年    8 月 2 1 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



出証番号    出証特 2 0 0 3 - 3 0 6 8 4 7 5

【書類名】 特許願

【整理番号】 HI030094

【提出日】 平成15年 2月28日

【あて先】 特許庁長官 殿

【国際特許分類】 H04L 12/00

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 本田 政則

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 平田 俊明

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 三宅 滋

【発明者】

【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

【氏名】 泉田 裕典

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 黒田 沢希

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 村瀬 敦史

## 【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 井上 太郎

## 【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

【氏名】 木原 健一

## 【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

## 【代理人】

【識別番号】 100071283

## 【弁理士】

【氏名又は名称】 一色 健輔

## 【選任した代理人】

【識別番号】 100084906

## 【弁理士】

【氏名又は名称】 原島 典孝

## 【選任した代理人】

【識別番号】 100098523

## 【弁理士】

【氏名又は名称】 黒川 恵

## 【選任した代理人】

【識別番号】 100112748

## 【弁理士】

【氏名又は名称】 吉田 浩二

## 【選任した代理人】

【識別番号】 100110009

【弁理士】

【氏名又は名称】 青木 康

## 【手数料の表示】

【予納台帳番号】 011785

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ジョブ管理方法、情報処理システム、プログラム、及び記録媒体

【特許請求の範囲】

【請求項 1】 情報処理装置を含んで構成される情報処理システムにおけるジョブ管理方法において、

ジョブ定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とを記憶するステップと、

前記ジョブ定義文の雛形と、前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とに基づいて、前記ユーザインタフェースによりユーザが設定する内容に基づいてジョブ定義文を生成する処理を実行させるためのデータを生成するステップと、

生成された前記データに基づいて前記処理を実行することによりジョブ定義文を生成するステップと、

を備えることを特徴とするジョブ管理方法。

【請求項 2】 前記ジョブ定義文の雛形は、XML 形式で記述されていること、を特徴とする請求項 1 に記載のジョブ管理方法。

【請求項 3】 前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形は、前記ジョブ定義文の雛形が記述されるファイルと同じファイルに XML 形式で記述されていること、を特徴とする請求項 2 に記載のジョブ管理方法。

【請求項 4】 前記ユーザインタフェースには、ユーザに設定を促すように構成された画面を表示してユーザに前記ジョブ定義文の設定を促す機能が含まれること、を特徴とする請求項 1 に記載のジョブ管理方法。

【請求項 5】 前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータに、ユーザに設定させる項目を前記画面に表示させるかどうかを制御するためのデータが記述されていること、

を特徴とする請求項 4 に記載のジョブ管理方法。

【請求項 6】 前記ジョブ定義文を設定させるためのユーザインタフェース

を規定するデータに、先行して表示される画面に対してユーザが設定した内容に応じてこれに後行して表示するように規定されている画面を表示するかどうかを制御するためのデータが記述されていること、を特徴とする請求項 4 に記載のジョブ管理方法。

【請求項 7】 前記ジョブ定義文の雛形には、前記ユーザインタフェースによりユーザに設定された内容に基づいて前記雛形に記述されている特定の記述部分を無効化させるための定義文が含まれていること、を特徴とする請求項 1 に記載のジョブの管理方法。

【請求項 8】 前記ジョブ定義文の雛形には、前記ユーザインタフェースによりユーザに設定された内容に基づいて前記雛形に記述されている特定の記述部分を繰り返し記述したジョブ定義文を生成させるように指示する定義文が含まれていること、を特徴とする請求項 1 に記載のジョブ管理方法。

【請求項 9】 前記ジョブ定義文には、複数のジョブを一連の流れとして実行させるジョブネットワークを設定するための定義文が含まれること、を特徴とする請求項 1 に記載のジョブ管理方法。

【請求項 10】 ジョブ定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とを記憶する手段と、

前記ジョブ定義文の雛形と、前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とに基づいて、前記ユーザインタフェースによりユーザが設定する内容に基づいてジョブ定義文を生成する処理を実行させるためのデータを生成する手段と、

生成された前記データに基づいて前記処理を実行することによりジョブ定義文を生成する手段と、

を備えることを特徴とする情報処理装置を含んで構成される情報処理システム。

【請求項 11】 ファイル形式のデータとして管理されている前記ジョブ定義文の雛形をインポートもしくはエクスポートする手段を備えること、を特徴とする請求項 10 に記載の情報処理システム。

【請求項 12】 情報処理装置を含んで構成される情報処理システムに、

ジョブ定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とを記憶する機能と、

前記ジョブ定義文の雛形と、前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とに基づいて、前記ユーザインタフェースによりユーザが設定する内容に基づいてジョブ定義文を生成する処理を実行させるためのデータを生成する機能と、

生成された前記データに基づいて前記処理を実行することによりジョブ定義文を生成する機能と、  
を実現させるためのプログラム。

【請求項 1 3】 情報処理装置に、ジョブの定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とに基づいて、前記ユーザインタフェースによりユーザが設定する内容に基づいてジョブの定義文を生成する処理を実行させるためのデータを生成する機能を実現させるためのプログラム。

【請求項 1 4】 ジョブの定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とを記憶する手段と、前記ジョブ定義文の雛形と前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とに基づいて前記ユーザインタフェースによりユーザが設定する内容に基づいてジョブ定義文を生成する処理を実行させるためのデータを生成する手段と、を備える情報処理システムにおける情報処理装置に、

生成される前記データに基づいて前記ユーザインタフェースを実現し、前記ジョブ定義文を生成する機能を実現するためのプログラム。

【請求項 1 5】 情報処理装置を含んで構成される情報処理システムに、ジョブ定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とを記憶する機能と、

前記ジョブ定義文の雛形と、前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とに基づいて、前記ユーザインタフェースによりユーザが設定する内容に基づいてジョブ定義文を生成する処理を実行させるためのデータを生成する機能と、

生成された前記データに基づいて前記処理を実行することによりジョブ定義文を生成する機能と、

を実現させるためのプログラムを記録した記録媒体。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

本発明は、ジョブ管理方法、情報処理システム、プログラム、及び記録媒体に関する。

##### 【0002】

#### 【従来の技術】

情報処理装置であるコンピュータを含んで構成される情報処理システムにおいて、コンピュータに実行させるジョブの設定は、一般にエディタを用いてジョブ制御言語を記述したりジョブの設定のためのツールを用いることにより行われている。またG U I (Graphical User Interface)を用いてジョブの設定を行えるようにしているもの存在する。

##### 【0003】

#### 【発明が解決しようとする課題】

ところで、情報技術の発達した今日では、ジョブの設定を行うためにより専門的かつ複雑な技術が要求されるようになってきている。とくに大規模かつ複雑な構成からなる情報処理システムにおけるジョブの設定に際しては、C P U、メモリ、ディスク、バックアップ装置等の資源の利用状況などを考慮して効率よくジョブを設定することが要求される。また、ジョブ間を連携させて一連のストリームとしてジョブを実行させるジョブネットワークの定義に際しては、ジョブ間の連携をスムーズに進行させるために高度な技術が要求される。このようにジョブ設定を行うユーザやオペレータ等の作業負担が増しており、コンピュータシステムにおいて実行させるジョブを効率よく管理するための仕組みが求められている。

##### 【0004】

この発明は、ジョブ管理方法、情報処理システム、プログラム、及び記録媒体



を提供することを目的とする。

#### 【0005】

##### 【課題を解決するための手段】

この目的を達成するための本発明のうち主たる発明は、  
情報処理装置を含んで構成される情報処理システムにおけるジョブ管理方法において、

ジョブ定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とを記憶するステップと、

前記ジョブ定義文の雛形と、前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とに基づいて、前記ユーザインタフェースによりユーザが設定する内容に基づいてジョブ定義文を生成する処理を実行させるためのデータを生成するステップと、

生成された前記データに基づいて前記処理を実行することによりジョブ定義文を生成するステップと、を備えることとする。

#### 【0006】

ジョブ定義文は、例えば、後述するポリシー定義XMLに記述されている。ジョブ定義文の雛形は、例えば、後述するポリシーテンプレート定義XMLに記述されている。ジョブ定義文を設定させるためのユーザインタフェースを規定するデータとは、例えば、後述するポリシーテンプレート定義の要素（エレメント）であるウィザードページである。ウィザードページの雛形は、例えば、後述するポリシーテンプレート定義XMLに記述される。ユーザインタフェースとは、例えば、ポリシーテンプレート定義のウィザードページの記述に基づいて、後述するポリシーウィザードGUIが生成する設定誘導画面（ウィザード画面）である。なお、ユーザインタフェースは、このように画面によるもののほか、音声によるものなど視覚以外の五感を用いるものであってもよい。

#### 【0007】

この発明によれば、ジョブ定義文の雛形に加えて、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形が用意されている。この雛形を利用することで、例えば、ユーザやオペレータ等にジョブ定義文を生成させ

る際のユーザインタフェース（例えば、設定誘導画面（ウィザード画面））を効率よく生成することができる。また、情報処理システムの構成、要素、ユーザニーズ、適用される業務等に応じて、柔軟なユーザインタフェースを提供することが可能となり、ジョブ定義文の設定にかかるユーザやオペレータ等の作業負担が軽減される。

#### 【0008】

なお、本発明の他の特徴については、本明細書及び添付図面の記載により明らかにする。

#### 【0009】

##### 【発明の実施の形態】

図1に本発明の一実施例として説明する情報処理システムのハードウェア構成を示している。情報処理システムは、CPU111、ROM・RAMなどのメモリ112からなるコンピュータ110と、ディスプレイなどの表示装置120、キーボード及びマウスなどの入力インタフェース130、ハードディスク装置・CD-ROM装置・DVD-ROM装置等の外部記憶装置140等を含んでいる。コンピュータ110は、メインフレームコンピュータ、ワークステーション、パーソナルコンピュータ等である。コンピュータ110上では、例えばUNIX（登録商標）などのオペレーティングシステムが動作している。オペレーティングシステム上では各種のアプリケーションプログラムが動作している。また、オペレーティングシステム上では、後述する設定誘導画面が記載されたWebページを提供するWebサーバが動作している。外部記憶装置140に記憶されるデータは、オペレーティングシステムが有するファイル・システムの機能によりファイルを単位として管理されている。

#### 【0010】

本実施例において、ジョブとは、コンピュータ110に行わせようとする、コンピュータ110のユーザから見たひとまとまりの処理の単位をいう。本実施例の情報処理システムでは、ジョブを自動的に実行する仕組みを提供するジョブ管理システムが動作する。ジョブ管理システムは、コンピュータ110で実行されるプログラムにより実現される。ジョブ管理システムは、例えば、銀行のオンラ

インシステムの場合には、オンラインで収集したトランザクションデータの集計、分析、データのバックアップなどを行うバッチ処理のシステムである。ジョブ管理システムは、入力インタフェース 1 3 0 からユーザやオペレータ等によるジョブの設定および条件を受け付けて、これにより設定されたジョブを設定された条件に従って実行する。前記条件としては、ジョブを実行させる日時を指定するスケジュールに関する条件等がある。また、複数のジョブを一連の流れとして実行させるいわゆるジョブネットワークの設定では、先行ジョブの処理の実行状態（処理中、処理済等）に応じた後行ジョブの実行条件など、ジョブ間の制約に基づく条件がある。

#### 【0 0 1 1】

図 2 はジョブ管理システムの構成および仕組みを説明している。ジョブ管理システムは、ユーザやオペレータ等によりジョブの定義文（以下、ポリシールール (Policy Rule) と称する）を設定させるためのユーザインタフェース 2 2 0 を提供する。ジョブ管理システムは、上記ユーザインタフェースにより設定されたポリシールールを XML (eXtensible Markup Language) 形式のファイルとしてコンピュータ 1 1 0 のメモリ 1 1 2 や外部記憶装置 1 4 0 に記憶する。なお、ポリシールールが記述された XML ファイルのことを、以下、ポリシー定義 XML 2 1 0 と称する。

#### 【0 0 1 2】

上記ユーザインタフェース 2 2 0 は、主として図 2 におけるポリシーウィザード GUI (Policy Wizard GUI) 2 3 0 の機能によって実現される。ジョブ管理システムに関する各種の情報が登録されて管理されているデータベースであるリポジトリ 2 4 0 (Repository) には、ポリシールールの原型となるデータとユーザインタフェース 2 2 0 を規定するデータとを含んだポリシーテンプレート (Policy Template) 定義 2 5 0 が登録されている。

#### 【0 0 1 3】

ポリシーウィザード GUI 2 3 0 は、リポジトリ 2 4 0 に登録されているポリシーテンプレート定義 2 5 0 に従い、上記ユーザインタフェース 2 2 0 として、ジョブの定義文の設定をユーザに促すように構成された一連の画面である設定誘

導画面（ウィザード（Wizard）画面）が記載された Web ページを生成し、生成した設定誘導画面を順次表示する。ポリシーテンプレート定義 250 は、ユーザインタフェース 220 によりユーザが設定する内容に基づいてジョブ定義文を生成する処理を実行させるためのデータである。ポリシーウィザード GUI 230 は、この設定誘導画面に従ってユーザが設定した内容を前記ポリシールールの原型となるデータに反映させていくことにより、ポリシールールを生成する。

#### 【0014】

ポリシーウィザード GUI 230 によって生成されたポリシールールは、図 2 に示すポリシー実行エンジン 260 によって解釈されて、ポリシー（Policy）定義情報・実行時間情報（スケジュール情報）270 としてリポジトリ 240 に登録される。ポリシー実行エンジン 260 は、リポジトリ 240 に登録されているポリシー定義情報・実行時間情報 270 の内容に基づいてジョブの実行を制御する。

#### 【0015】

図 2 に示されているポリシーコントロール GUI（Policy Control GUI）280 は、ユーザやオペレータ等が、リポジトリ 240 の管理やポリシー実行エンジン 260 の管理を行うための GUI である。ユーザやオペレータ等は、このポリシーコントロール GUI 280 を利用して、ポリシールールの参照、削除、有効化、無効化などを行うことができる。ポリシーコントロール GUI 280 を利用することで、ユーザやオペレータ等は既に設定されているポリシールールを事後的に管理することができる。

#### 【0016】

上記ポリシーテンプレート定義 250 は、図 2 におけるポリシーテンプレートパーサ（Policy Template Parser）290 によりリポジトリ 240 に登録される。図 2 におけるポリシーテンプレート（Policy Template）定義 XML 292 は、ポリシーテンプレートパーサ 290 がポリシーテンプレート定義 250 をリポジトリ 240 に登録する際に参照する。ポリシーテンプレート定義 XML 292 には、ジョブ定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とが XML 形式で記述されている。

**【0017】**

ユーザやオペレータ等は、編集したポリシーテンプレート定義XML 292をポリシーテンプレートパーサ290の実行に際しポリシーテンプレートパーサ290に引き渡す。この引渡しは、例えば、ポリシーテンプレート定義XML 292のファイル名を、ポリシーテンプレートパーサ290の実行コマンドに引数として指定することで行われる。ポリシーテンプレートパーサ290は、引き渡されたポリシーテンプレート定義XML 292の内容に従ってポリシーテンプレート定義250を生成し、これをリポジトリ240に登録する。

**【0018】**

このように、ポリシーテンプレート定義XML 292を編集することにより、ポリシーテンプレート定義250の生成とそのリポジトリ240への登録作業を効率よく行うことができる。また、生成するポリシールールの内容に応じた内容のジョブ定義文の雛形とジョブ定義文を設定させるためのユーザインタフェース220を規定するデータの雛形とが記述されたポリシーテンプレート定義XML 292とがあらかじめ用意されていることで、ポリシーテンプレート定義250を効率よく生成することができる。従って、情報処理システムの管理者等は、ユーザやオペレータ等にジョブ定義文を生成させる際のユーザインタフェース（例えば、設定誘導画面（ウィザード画面））を効率よく生成することができる。

**【0019】**

このようにこれらの雛形を用いてポリシーテンプレート定義XML 292の編集が行われることで、ジョブ管理システムの管理者等にかかる作業負担が軽減され、効率よくポリシーテンプレート定義250を生成することができる。また情報処理システムの構成、要素、ユーザニーズ、適用される業務等に応じて、様々なタイプのユーザインタフェースを柔軟に提供することが可能となり、ジョブ定義文の設定にかかるユーザやオペレータ等の作業負担が軽減される。

**【0020】**

===ポリシールール===

ポリシールールは、上述したようにXML形式のファイルであるポリシー定義XML 210として管理される。このようにポリシールールがXML形式のファ

イルとして管理されることで、例えば、他の情報処理システムとの間でのポリシーールのやりとりを簡便に行うことができる。これにより他の情報処理システムが保有する既存の資産を有効に活用することができ、また他の情報処理システムに資産を供給することができる。

#### 【0021】

また、ポリシーールの記述文法としてXMLを用いることで、例えば、①タグの名称から画面との対応関係を容易に把握することができる、②情報内容に固有の名称をつけることができる、③タグや要素（エレメント）の相互関係を自由に決めることができる、といったXMLが本来的に有している優れた特徴が生かされ、また新規に規定されるXMLの仕様を取り込んでいくことで将来的な拡張性も確保される。

#### 【0022】

ポリシー定義XML 210の記述文法は基本的にXMLの記述文法に従うが、本実施例におけるジョブ管理システムは、このようなXMLの基本的な記述文法に加え、さらに各種の要素（エレメント）を解釈し対応する処理を実行する機能を備えている。本発明の情報処理システムに特有のエレメントのうち、主なものを図3に例示している。この図において、ポリシー（Policy）は、ポリシー定義XML 210のルート・エレメントである。ポリシーグループ（Policy Group）は、ポリシーグループ（Policy Group）や後述するルートポリシーール（Policy Rule）をグルーピングするエレメントである。ルートポリシーール（Policy Rule）は、後述するポリシーコンディション（Policy Condition）、ポリシーアクション（Policy Action）、および上記ポリシーール（Policy Rule）をグルーピングするエレメントである。

#### 【0023】

ポリシーウェイトイベントコンディション（Policy Wait Event Condition）は、イベントを待つ場合に記述されるエレメントである。このエレメントは、イベントドリブン型のポリシーを記述したい場合に用いられるエレメントである。ポリシースケジュールコンディション（Policy Schedule Condition）は、指定時刻を待つ場合に記述されるエレメントである。このエレメントは、後述するポ

リシーコマンドアクション (Policy Command Action) を、決まった時刻に実行したい場合に記述されるエレメントである。ポリシーコマンドリザルトコンディション (Policy Command Result Condition) は、指定したポリシーコマンドアクション (Policy Command Action) の終了を待つ場合に記述されるエレメントである。ポリシーコマンドアクション (Policy Command Action) の実行結果が指定した値であれば、兄弟エレメントであるポリシーコマンドアクション (Policy Command Action) を実行する。

#### 【 0 0 2 4 】

ポリシールールバリディティペリオッド (Policy Rule Validity Period) は、ルートポリシールール (Policy Rule) の有効期間を定義する場合に記述されるエレメントである。コンパウンドポリシーアクション (Compound Policy Action) は、ポリシーアクション (Policy Action) をグルーピングするエレメントである。また、ポリシーコマンドアクション (Policy Command Action) は、兄弟エレメントであるポリシーウェイトイベントコンディション (Policy Wait Event Condition) や、ポリシースケジュールコンディション (Policy Schedule Condition) 、ポリシーコマンドリザルトコンディション (Policy Command Result Condition) 等の条件が成立したときに実行されるコマンドである。

#### 【 0 0 2 5 】

ポリシー定義XML 2 1 0 には、あるルートのポリシールール (Policy Rule) の下位エレメントとして記述されている全てのポリシーコマンドアクション (Policy Command Action) が共通に参照することができる環境変数である、いわゆるグローバル環境変数を定義することができる。複数のコンピュータを含んだ情報処理システムにおいては、ポリシーコマンドアクション (Policy Command Action) を実行するコンピュータが必ずしも同じコンピュータであるとは限らない。つまりグローバル環境変数は、複数のコンピュータが対象となるジョブが設定されている場合に各ポリシーコマンドアクション (Policy Command Action) で共通に参照させる変数を持たせたい場合に有用である。なお、各ポリシーコマンドアクション (Policy Command Action) でグローバル環境変数の初期値を設定することができる。またポリシーグループ (Policy Group) とルートポリシー

ルール (Policy Rule) にグローバル環境変数の初期値を設定することもできる。

#### 【0 0 2 6】

ポリシー定義XML 2 1 0 には、以上に説明したエレメントがXMLの文法に従って記述される。つまり、ポリシーウィザードGUIは、リポジトリ2 4 0 に登録されているポリシーテンプレート定義に従って、ユーザインタフェース2 2 0 に対してユーザやオペレータ等が設定した内容を、ポリシーテンプレート定義に含まれている前記ポリシールールの原型となるデータに反映させることにより、ポリシー定義XML 2 1 0 を生成する。生成されたポリシー定義XML 2 1 0 の一例としてそのDTD (Document Type Definition) 部分のサンプルを図4に示す。

#### 【0 0 2 7】

===ポリシーテンプレート定義===

ポリシーテンプレート定義2 5 0 は、上述したようにポリシールールの原型となるデータと上記ユーザインタフェース2 2 0 を規定するデータとを記述したものである。ポリシーテンプレート定義2 5 0 には、ポリシーウィザードGUI 2 3 0 が、ポリシールールを設定するためにディスプレイに表示する上記設定誘導画面や、生成されるポリシー定義XML 2 1 0 の原型となる情報が含まれている。

#### 【0 0 2 8】

ポリシーテンプレート定義2 5 0 の構成を図5に示している。またポリシーテンプレート定義2 5 0 の主要素 (エレメント) を図6に示している。図5または図6において、ポリシーテンプレートセット (Policy Template Set) は、全ての要素のルート要素である。ポリシーテンプレートグループ (Policy Template Group) は、ポリシーテンプレートをグルーピングするための要素である。なお、ポリシーテンプレートグループは、入れ子に記述することが可能である。ポリシーテンプレートグループは、ファイル・システムのフォルダ (ディレクトリ) の様な役割を担う。ポリシーテンプレート (Policy Template) は、1 つのポリシールールを生成するための各種の定義を行うための要素である。



**【 0 0 2 9 】**

ポリシーステンシル (Policy Stencil) は、ポリシールール (ポリシー定義 XML) の原型を定義する要素である。ポリシーステンシルには、つぎのようなステートメントを記述することができる。例えば、式が真ならば、開始ステートメントと終了ステートメントに囲まれた記述部分が有効な記述としてポリシー定義 XML 2 1 0 に書き出されるが、式が偽であればポリシー定義 XML 2 1 0 には書き出されないように制御するステートメントを記述することができる。図 7 にそのようなステートメント (この図ではこのステートメントは「@Validf」というコマンドである) の一例を示す。また、例えば、リストの要素を順に変数に代入し、全ての要素について開始ステートメントと終了ステートメントに挟まれた記述部分を繰り返しポリシー定義 XML 2 1 0 に書き出すように制御するステートメントを記述することができる。図 8 にそのようなステートメント (この図ではこのステートメントは「@ForEach」というコマンドである) の一例を示す。

**【 0 0 3 0 】**

ウィザードページ (Wizard Page) は、設定誘導画面 (ウィザード画面) を定義するための要素である。1 つのウィザードページはウィザード画面の 1 ページ (一枚の Web ページ) に対応する。ウィザードページには、ユーザに設定させる項目を前記画面に表示させるかどうかを制御するためのデータを記述することができる。また、先行して表示される設定誘導画面に対してユーザが設定した内容に応じてこれに後行して表示するように規定されている設定誘導画面を、表示するかどうかを制御するためのデータを記述することができる。

**【 0 0 3 1 】**

1 つのウィザードページには、複数 (1 つでもよい) のパラメタ設定コントロールを定義することができる。パラメタ設定コントロールは、ポリシーテンプレートバリアブル (Policy Template Variable) を確定 (値の代入) するための要素である。1 つのパラメタ設定コントロールは、1 つのポリシーテンプレートバリアブルを確定する。パラメタ設定コントロールには、テキストライン (Text Line)、テキストエリア (Text Area)、セレクトアイテム (Select Item) の 3 つのコントロールが含まれる。このうちテキストラインは 1 行の文字列を指定する

ためのコントロールである。テキストエリアは、複数行の文字列を指定するためのコントロールである。セレクトアイテムは、複数の項目を表示し、特定の項目を選択するためのコントロールである。ポリシーテンプレートバリアブル (Policy Template Variable) は、ポリシーテンプレートで宣言される変数であり、ウィザードページ内のパラメタ設定コントロールで値の確定方法が定義され、ポリシーシステム内またはウィザードページ内で値が参照される。リソースタイプ (Resource Type) は、ポリシーテンプレートがどのリソースと関連付けられるかを定義するための要素である。

### 【 0 0 3 2 】

===ポリシーテンプレート定義XML等===

ポリシーテンプレート定義XML 2 9 2 は、上述したようにポリシーテンプレート定義 2 5 0 を生成するためのデータがXMLに従って記述されているファイルである。ポリシーテンプレート定義XML 2 9 2 は、その内部にポリシーシステムの形で上記のポリシー定義XML 2 1 0 の雛形となるXML文を含んでいる。上述のポリシーテンプレート定義 2 5 0 における上記ウィザードページを規定するデータの雛形も、ポリシーテンプレート定義XML 2 9 2 に、ジョブ定義文の雛形とともにXML形式で記述されている。図 9 にポリシーテンプレート定義XML 2 9 2 の一例としてそのDTD部分のサンプルを示す。

### 【 0 0 3 3 】

ポリシーテンプレートパーサ 2 9 0 は、入力されるポリシーテンプレート定義XML 2 9 2 を解釈することにより、ポリシーテンプレート定義XML 2 9 2 に従ったポリシーテンプレート定義 2 5 0 をリポジトリ 2 4 0 に登録する。ポリシーテンプレートパーサ 2 9 0 は、例えば J a v a (登録商標) クラスとしてこの情報処理システムに実装されている。ポリシーテンプレートパーサ 2 9 0 は、必要な場合にはコマンドラインからも起動できるように C L I (Command Line Interface) の機能も提供する。

### 【 0 0 3 4 】

ポリシーテンプレートパーサ 2 9 0 が、ポリシーテンプレート定義XML 2 9 2 に基づいてポリシーテンプレート定義 2 5 0 をリポジトリ 2 4 0 に登録する機

能（インポート機能）について説明する。例えば入力インタフェース 1 3 0 によりインポート機能の実行指示が発生すると、ポリシーテンプレートパーサ 2 9 0 は、ポリシーテンプレート定義 XML 2 9 2 を読み込んでこれを解釈する。この解釈に際し、内容的に問題が無ければポリシーテンプレート定義 XML 2 9 2 に基づいて生成したポリシーテンプレート定義 2 5 0 をリポジトリ 2 4 0 に登録する。

#### 【 0 0 3 5 】

図 1 0 はポリシーテンプレートパーサ 2 9 0 によるインポート機能に関する処理を説明する P A D (Problem Analysis Diagram) である。インポート機能の実行指示が発生すると、ポリシーテンプレートパーサ 2 9 0 は、まず変数のクリアや初期値の設定などの初期処理を行い (S1011)、また、インポート機能の実行指示命令に付帯指定されている引数の解析 (S1012) を行う。次にポリシーテンプレートパーサ 2 9 0 は、リポジトリ 2 4 0 の内容をチェックし (S1013)、リポジトリ 2 4 0 の内容と引数の内容との間に矛盾などの問題がないかどうかを調べる (S1014)。ここで問題があれば処理を中断し例外処理を実行する (S1015)。一方、問題が無ければ、ポリシーテンプレートパーサ 2 9 0 は、引数として指定されているファイル名に対応するポリシーテンプレート定義 XML 2 9 2 の内容を解析し、このポリシーテンプレート定義 XML 2 9 2 に対応するポリシーテンプレート定義 2 5 0 をリポジトリ 2 4 0 に登録する (S1016)。以上のようにしてインポート機能に関する処理が行われる (S1017)。

#### 【 0 0 3 6 】

なお、本実施例のジョブ管理システムにおけるポリシーテンプレートパーサ 2 9 0 は、上述のインポート機能以外にも、リポジトリ 2 4 0 に登録されているポリシーテンプレート定義 2 5 0 に基づいてポリシーテンプレート定義 XML 2 9 2 を生成するエクスポート機能や、リポジトリ 2 4 0 に登録されているポリシーテンプレート定義 2 5 0 を削除する機能等を備えている。

#### 【 0 0 3 7 】

=== ポリシールールの設定 ===

ユーザやオペレータ等は、ポリシーウィザード G U I 2 3 0 が表示する設定誘

導画面に従ってポリシールールを作成する。図 1 1 はポリシーウィザード G U I 2 3 0 によりポリシールールが生成される処理を説明する P A D である。ポリシーウィザード G U I 2 3 0 は、変数のクリアや初期値の設定などの初期処理を行い (S1111)、インポート機能の実行指示命令に付帯指定されている引数の解析 (S1112) を行う。次にポリシーウィザード G U I 2 3 0 は、リポジトリ 2 4 0 をチェックし (S1113)、リポジトリ 2 4 0 の登録内容と引数の内容との間の矛盾などの問題がないかどうかを調べる (S1114)。ここで問題があれば処理を中断し例外処理を実行する (S1115)。次にポリシーウィザード G U I 2 3 0 は、ポリシーテンプレート要素の属性、ポリシーテンプレートバリアブル要素の属性、及びポリシーステンシル要素の値をリポジトリ 2 4 0 から読み出す (S1116~S1118)。その後、ポリシーウィザード G U I 2 3 0 はポリシーテンプレート定義 2 5 0 に含まれる各ウィザードページ要素についての処理を開始する (S1119)。

#### 【 0 0 3 8 】

図 1 2 はウィザードページ要素についての処理を説明する P A D である。ポリシーウィザード G U I 2 3 0 は、まずウィザードページ要素の属性をリポジトリ 2 4 0 から読み出す (S1211)。またこの際には、ポリシーテンプレートバリアブル (Policy Template Variable) の評価も行う。次にポリシーウィザード G U I 2 3 0 は、スキップするページであるかどうかを判定し (図ではスキップルールの評価) (S1212)、スキップするページである場合には (S1213)、当該ウィザードページ要素についての処理を終了する (S1214)。また、ポリシーウィザード G U I 2 3 0 は、次に処理すべきウィザードページ要素が存在する場合にはその処理を開始し、存在しない場合には (S1220) の処理に進む。

#### 【 0 0 3 9 】

(S1215) の処理では、ポリシーウィザード G U I 2 3 0 は、当該ウィザードページの要素に含まれる各パラメタ設定コントロールについての処理を開始する。まずポリシーウィザード G U I 2 3 0 は、パラメタ設定コントロールの属性をリポジトリ 2 4 0 から読み出す。ここではポリシーテンプレートバリアブ

ルの評価も行う (S1216)。なお、ここでいうパラメタ設定コントロールとは、「TextLine」、「TextArea」、「SelectItem」のうちのいずれかである。また、パラメタ設定コントロールに子エレメントが存在する場合には、その属性を読み出して評価する。またこの際には、ポリシーテンプレートバリアブルの評価も行う (S1217)。

#### 【 0 0 4 0 】

続く (S1218) の処理では、読み出された属性などに基づいて表示装置 1 2 0 に設定誘導画面 (ウィザード画面) を表示する。ユーザはこの設定誘導画面に対してパラメタを設定する。ユーザは設定が終われば次ページボタン (Next ボタン) をクリックするなどして次の画面に進むための操作を行う (S1219)。なお、この実施例では、比較的単純なユーザオペレーションを想定した説明としているが、表示される設定誘導画面の構成や表示方法はこれに限定されるものではない。

#### 【 0 0 4 1 】

図 1 1 における (S1119) の処理において、全てのウィザードページエレメントについての (S1200) の処理が完了すると、次にポリシーウィザード G U I 2 3 0 はパラメタ確認画面を表示する (S1120)。ユーザやオペレータ等は、設定誘導画面に表示されるパラメタ確認画面上でパラメタの確認やポリシールールの設定などを行い、その結果問題がなければ完了ボタン (Finish ボタン) をクリックする (S1121)。次にポリシーウィザード G U I 2 3 0 は、ポリシーステンシルにポリシーテンプレートバリアブルの値を代入する (S1122)。またポリシーウィザード G U I 2 3 0 は、ポリシーステンシル制御ステートメントの評価を行って、ポリシールールを完成させる (S1123)。

#### 【 0 0 4 2 】

完成したポリシールールは、ポリシー定義 XML 2 1 0 として出力されて、ポリシー実行エンジン 2 6 0 に引き渡される (S1124)。この引渡しは図示しないポリシーパーサの機能によって行われる。ポリシーパーサは、完成したポリシー定義 XML 2 1 0 を解釈してポリシールールのインスタンスを生成し、ポリシー実行エンジン 2 6 0 に登録する。以上の処理が完了すると、ポリシーウィザード

G U I 2 3 0 は、ポリシールール生成完了画面を表示して (S1125)、ポリシールール (ポリシー定義XML) の生成処理を終了する (S1126)。

#### 【0 0 4 3】

なお、リポジトリ 2 4 0 に複数のポリシーテンプレート定義 2 5 0 が登録されている場合には、図 1 1 に示す処理は登録されているポリシーテンプレート定義 2 5 0 ごとに行われる。そしてこのようにリポジトリ 2 4 0 に複数のポリシーテンプレート定義 2 5 0 が登録されている場合には、各ポリシーテンプレート定義 2 5 0 に対応した設定誘導画面が提供されることになる。

#### 【0 0 4 4】

=== 応用例 ===

次に、以上に説明したジョブ管理システムが動作する情報処理システムの応用例について説明する。図 1 3 に示す情報処理システムは、例えば、企業のシステムセンターや、私的もしくは公的に運営されるデータセンタなどで運用されている典型的なストレージシステムである。このストレージシステムにおいて、ジョブ管理システムを実現するためのプログラムは、管理サーバ 1 3 2 0 で動作している。管理サーバ 1 3 2 0 は、このストレージシステムの保守や運用、管理等を行うためのコンピュータである。管理サーバ 1 3 2 0 は、例えば、ストレージ装置 1 3 0 0 を業務サーバ 1 3 1 0 から使用できるようにするための設定、SAN の設定、業務サーバ 1 3 1 0 にバッチ処理を実行させるための設定、ストレージ装置 1 3 0 0 に記憶されるデータのバックアップをテープデバイス 1 3 9 0 に記憶させる処理等を行う場合に利用される。

#### 【0 0 4 5】

オペレータ等が、管理サーバ 1 3 2 0 で動作するジョブ管理システムに対して情報処理システム内の各機器に行わせるべきジョブをポリシールールとして定義すると、このポリシールールに従った処理が各機器において行われるように、各機器を制御するコマンドが管理サーバ 1 3 2 0 により実行される。例えばオペレータ等が所定時刻にバックアップを実行するというポリシールールが設定されると、管理サーバ 1 3 2 0 は、このポリシールールに従った処理が行われるように後述の論理ボリューム 1 3 6 0 やテープデバイス 1 3 9 0 を制御するためのコマ

ンドを実行する。

#### 【0046】

管理サーバ（情報処理装置）1320は、管理LAN1341を介して業務サーバ1310、FC-SW1（1350）、ストレージ装置1300及びテープデバイス1390に接続されている。

#### 【0047】

管理サーバ1320の構成を図14に示している。管理サーバ1320は、CPU1410、メモリ1411、LAN I/F（LAN InterFace）1412、記録媒体読取装置1420、入力装置1425、出力装置1426、記憶装置1430を備える。記録媒体読取装置1420は、記録媒体1421に記録されているプログラムやデータを読み取るための装置である。例えば記録媒体1421に記録されている、上述のジョブ管理システムを実現するためのプログラム1450を、記録媒体読取装置1420を用いて上記記録媒体1421から読み取って、メモリ1411や記憶装置1430に格納するようにすることができる。記録媒体1421としては、フレキシブルディスクやCD-ROM、DVD-ROM、半導体メモリ等を用いることができる。

#### 【0048】

記憶装置1430は、例えばハードディスク装置やフレキシブルディスク装置、半導体記憶装置等である。入力装置1425はユーザやオペレータ等による管理サーバ1320へのデータの入力等のために用いられる。入力装置1425としては例えばキーボードやマウス等が用いられる。出力装置1426としては例えばディスプレイやプリンタ等が用いられる。LAN I/F 1412は業務サーバ1310やFC-SW1（1350）、ストレージ装置1300、テープデバイス1390と通信を行うための装置である。

#### 【0049】

業務クライアント1315は、基幹LAN（Local Area Network）1340を通じて業務サーバ1310に接続されている。業務サーバ1310及び業務クライアント1315は、CPU（Central Processing Unit）やメモリ、入出力装置等を備えたコンピュータである。これにより業務クライアント1315は業務

サーバ 1360 により提供される各種のサービスを受けることができる。

#### 【0050】

業務サーバ 1310 により提供されるサービスは、例えば銀行の自動預金預け払いサービスやインターネットのホームページ閲覧サービスのようなオンラインサービスを始め、科学技術分野における実験シミュレーションを行うバッチ処理サービス等である。

#### 【0051】

業務サーバ 1310 は、FC-SW1 (Fibre Channel-Switch1、ネットワーク装置) 1350 を介してストレージ装置 1300 及びテープデバイス (バックアップ装置) 1390 に接続されている。FC-SW1 (1350) は、業務サーバ 1310 とストレージ装置 1300 及びテープデバイス 1390 間を SAN (Storage Area Network) で接続するために用いられるスイッチである。

#### 【0052】

ストレージ装置 1300 は、業務サーバ 1310 が業務クライアント 1315 に情報処理サービスを提供する際に必要とする記憶資源を提供する。ストレージ装置 1300 は、例えばディスクアレイ装置である。記憶資源は、ストレージ装置 1300 が備えるディスクドライブにより提供される物理的な記憶領域上に論理的に設定される記憶領域である論理ボリューム 1360 として提供される。ディスクドライブとしては、例えばハードディスク装置やフレキシブルディスク装置、半導体記憶装置等様々なものを用いることができる。ディスクドライブは、RAID (Redundant Arrays of Inexpensive Disks) により管理されることもある。

#### 【0053】

以上本実施の形態について説明したが、上記実施例は本発明の理解を容易にするためのものであり、本発明を限定して解釈するためのものではない。本発明はその趣旨を逸脱することなく変更、改良され得ると共に、本発明にはその等価物も含まれる。

#### 【0054】

#### 【発明の効果】



本発明によれば、ジョブ管理方法、情報処理システム、プログラム、及び記録媒体を提供することができる。

**【図面の簡単な説明】**

【図 1】 本発明の一実施例による、情報処理システムのハードウェア構成を示す図である。

【図 2】 本発明の一実施例による、情報処理システムの構成および仕組みを説明する図である。

【図 3】 本発明の一実施例による、情報処理システムに特有のエレメントを示す図である。

【図 4】 本発明の一実施例による、ポリシー定義XMLのDTDのサンプルを示す図である。

【図 5】 本発明の一実施例による、ポリシーテンプレートの構成を示す図である。

【図 6】 本発明の一実施例による、ポリシーテンプレートの主要素（エレメント）を示す図である。

【図 7】 本発明の一実施例による、ポリシーステンシルに記述されるステートメントの一例を示す図である。

【図 8】 本発明の一実施例による、ポリシーステンシルに記述されるステートメントの一例を示す図である。

【図 9】 本発明の一実施例による、ポリシーテンプレート定義XMLのDTDのサンプルを示す図である。

【図 10】 本発明の一実施例による、ポリシーテンプレートパーサによるインポート機能に関する処理を説明するPADを示す図である。

【図 11】 本発明の一実施例による、ポリシーウィザードGUIによりポリシールールが生成される処理を説明するPADを示す図である。

【図 12】 本発明の一実施例による、ウィザードページエレメントについての処理を説明するPADを示す図である。

【図 13】 本発明の一実施例による、ジョブ管理システムが適用される情報処理システムの具体例を示す図である。

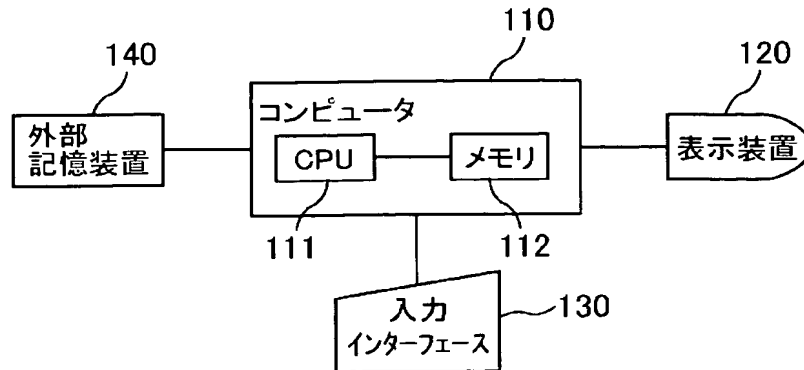
【図 1 4】 本発明の一実施例による、管理サーバの構成を示すブロック図を示す図である。

【符号の説明】

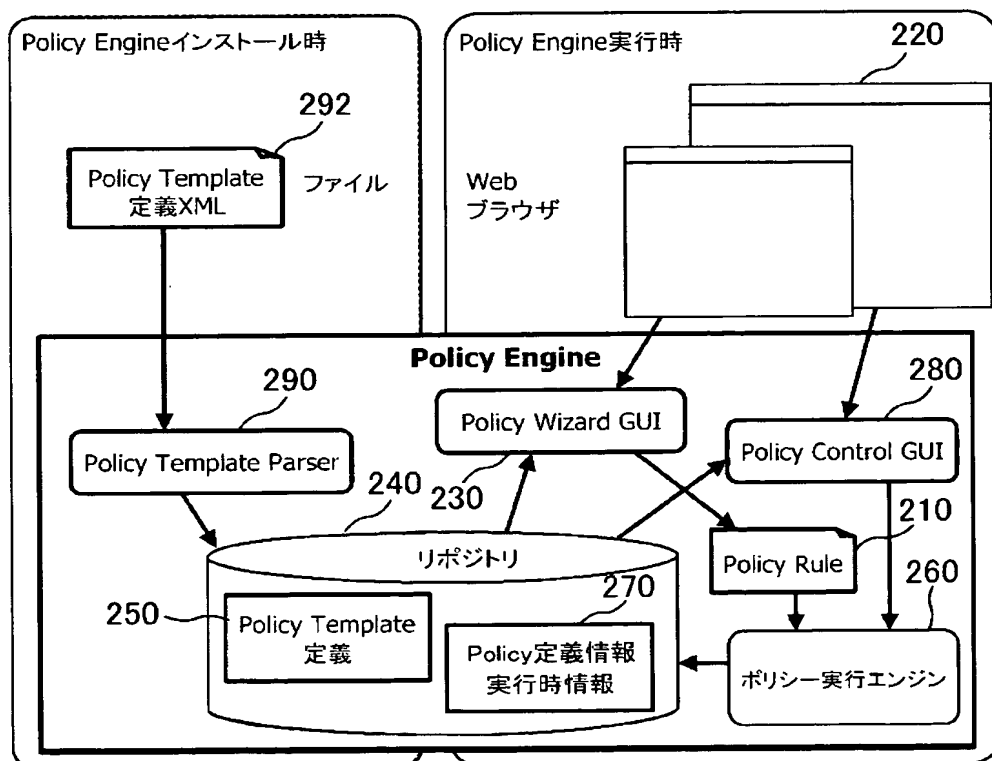
- 1 1 0 コンピュータ
- 1 2 0 表示装置
- 1 3 0 入力インタフェース
- 1 4 0 外部記憶装置
- 2 1 0 ポリシー定義XML
- 2 2 0 ユーザインタフェース
- 2 3 0 ポリシーウィザードGUI
- 2 4 0 リポジトリ
- 2 5 0 ポリシーテンプレート定義
- 2 6 0 ポリシー実行エンジン
- 2 7 0 ポリシー定義情報・実行時間情報
- 2 8 0 ポリシーコントロールGUI
- 2 9 0 ポリシーテンプレートパーサ
- 2 9 2 ポリシーテンプレート定義XML

【書類名】 図面

【図 1】



【図 2】

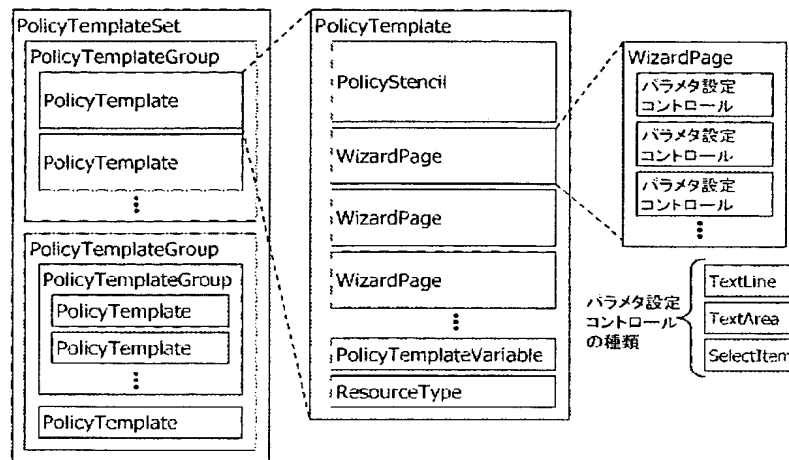


【図 3】

エレメント名	概 要
Policy	<ul style="list-style-type: none"> <li>Policy 定義 XML のルート・エレメント。</li> </ul>
PolicyGroup	<ul style="list-style-type: none"> <li>PolicyRule や PolicyGroup をグルーピングするエレメント。</li> <li>PolicyGroup は PolicyRule をツリー構造で整理するために利用される。</li> <li>カレンダー(休日)を定義する。</li> </ul>
PolicyRule	<ul style="list-style-type: none"> <li>PolicyCondition、PolicyAction、PolicyRule をグルーピングするエレメント。</li> <li>Policy または PolicyGroup の子エレメントである PolicyRule(つまり、上位の PolicyRule がない PolicyRule)をルート PolicyRule(または単に PolicyRule)と呼ぶ。</li> <li>PolicyRule の子エレメントである PolicyRule をサブ PolicyRule と呼ぶ。</li> <li>PolicyRule の子エレメントである PolicyCondition が成立すると、兄弟エレメントである PolicyAction とサブ PolicyRule が実行される。</li> </ul>
PolicyWaitEventCondition	<ul style="list-style-type: none"> <li>イベントを待つ。</li> <li>イベント・ドリブン型の Policy を記述したい場合に使う。</li> </ul>
PolicyScheduleCondition	<ul style="list-style-type: none"> <li>指定時刻を待つ。</li> <li>決まった時刻に PolicyAction を実行したい場合に使う。</li> </ul>
PolicyCommandResultCondition	<ul style="list-style-type: none"> <li>指定した PolicyCommandAction の終了を待つ。</li> <li>PolicyCommandAction の実行結果が指定した値であれば、兄弟エレメントである PolicyRule を実行する。</li> </ul>
PolicyRuleValidityPeriod	<ul style="list-style-type: none"> <li>ルート PolicyRule の有効期間を定義する。</li> </ul>
CompoundPolicyAction	<ul style="list-style-type: none"> <li>PolicyAction をグルーピングするエレメント。</li> </ul>
PolicyCommandAction	<ul style="list-style-type: none"> <li>兄弟エレメントである PolicyCondition が成立したときに実行されるコマンドを定義する。</li> </ul>



【図 5】



【図 6】

要素(エレメント)名	概 要
PolicyTemplateSet	以下の全ての要素のルート要素。
PolicyTemplateGroup	<ul style="list-style-type: none"> <li>PolicyTemplate をグルーピングするための要素。</li> <li>PolicyTemplateGroup は入れ子が可能であり、ファイル・システムのフォルダ(ディレクトリ)の様な役割を担う。</li> </ul>
PolicyTemplate	<ul style="list-style-type: none"> <li>1 つの Policy Rule を生成するための各種定義を行う要素。</li> <li>PolicyStencil、WizardPage、PolicyTemplateVariable、ResourceType などの子要素として持つ。</li> </ul>
PolicyStencil	<ul style="list-style-type: none"> <li>Policy Rule の原型を定義する要素。</li> <li>Policy 定義 XML で定義した PolicyRule 要素の内部に PolicyTemplateVariable の参照や Policy Stencil 制御ステートメントを含んだもの。</li> </ul>
WizardPage	<ul style="list-style-type: none"> <li>ウィザード画面(パラメタ設定ページ)を定義するための要素。</li> <li>1 つの WizardPage はウィザード画面の 1 ページとなる。</li> <li>1 つの WizardPage に複数(1 つでもよい)のパラメタ設定コントロールを定義可能。</li> </ul>
パラメタ設定コントロール	<ul style="list-style-type: none"> <li>PolicyTemplateVariable を確定(値の代入)するための要素。</li> <li>1 つのパラメタ設定コントロールは、1 つの PolicyTemplateVariable を確定する。</li> </ul>
TextLine	1 行の文字列を指定するためのコントロール。
TextArea	複数行の文字列を指定するためのコントロール。
SelectItem	複数の項目を表示し、特定の項目を選択するためのコントロール。
PolicyTemplateVariable	<ul style="list-style-type: none"> <li>PolicyTemplate で宣言される変数であり、WizardPage 内のパラメタ設定コントロールで値の確定方法が定義され、PolicyStencil 内または WizardPage 内で値が参照される。</li> </ul>
ResourceType	PolicyTemplate がどのリソースと関連付けられるかを定義するための要素。

【図 7】

```

    <@ ValidIf (式) [ @>    // 開始ステートメント
        Policy Ruleの記述
    ...
    <@ ] @>                // 終了ステートメント

```

・ここで、式とは以下の書式とする。ここで、値1や値2は一旦数値 (Integer) への変換が試みられ、失敗すると文字列として扱われる:

値1 = 値2	値1と値2が等しければ式は真
値1 != 値2	値1と値2が等しくなければ式は真
値1 <= 値2	値1が値2以下であれば式は真
値1 >= 値2	値1が値2以上であれば式は真
値1 < 値2	値1が値2より小さければ式は真
値1 > 値2	値1が値2より大きければ式は真

・なお式として複数の式を|| (論理和) や&& (論理積) で連結することが可能である。なお、優先順位を指定する括弧()も指定可能である。

【図 8】

```

    <@ ForEach 変数名 (リスト) [ @>    // 開始ステートメント
        Policy Ruleの記述
    ...
    <@ ] @>                // 終了ステートメント

```

・なお、開始ステートメントと終了ステートメントに挟まれた Policy Rule の記述の中には、以下のように「変数の参照」と「通し番号の参照」を記述できる。Policy Rule が生成されるとき、これらの参照は実際の値に置き換わる:

```

    <@= 変数名 @>
        ・リストの要素の値に置き換わる。
    <@= number (開始番号, 桁数) @>
        ・リストの要素の順番に対応した数値に置き換わる。
        ・開始番号と桁数が指定できる。例えば、開始番号が5で
          桁数が3の場合順に、"005", "006", "007"...という文字列
          に順に置き換わる。

```

【図 9】

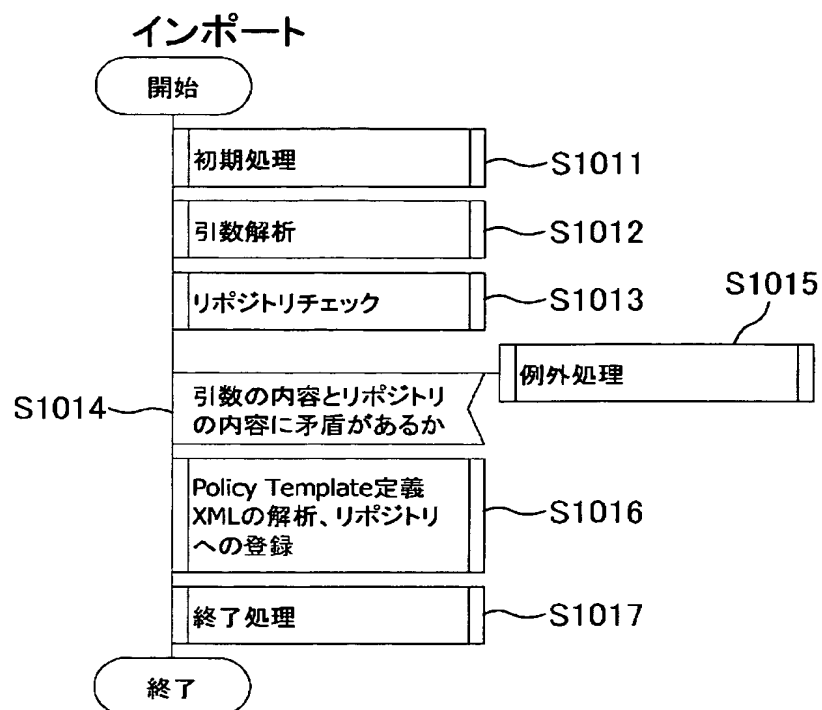
```

<!-- ***** -->
<!-- TextLine element -->
<!-- ***** -->
<ELEMENT TextLine
    (
        Command |
        TextFile |
        JavaClass
    )?
>
<ATTLIST TextLine
    SElementCommon:
    SAffix:
    SParamControlCommon:
    TextType
        (NormalText | Password)
    Size
        #IMPLIED
    MaxLength
        #IMPLIED
    DefaultValue
        #IMPLIED
    SParamControlStyle:
    SParamControlInputConstraint:
>
<!-- ***** -->
<!-- TextArea element -->
<!-- ***** -->
<ELEMENT TextArea
    (
        Command |
        TextFile |
        JavaClass
    )?
>
<ATTLIST TextArea
    SElementCommon:
    SAffix:
    SParamControlCommon:
    Rows
        #IMPLIED
    Columns
        #IMPLIED
    DefaultValue
        #IMPLIED
    SParamControlStyle:
>
DefaultPolicyRuleName
    #IMPLIED
WorkingPolicyGroupName
    #IMPLIED
>
<!-- ***** -->
<!-- PolicyTemplateVariable element -->
<!-- ***** -->
<ELEMENT PolicyTemplateVariable
    EMPTY
>
<ATTLIST PolicyTemplateVariable
    Name
        #REQUIRED
    IsArray
        (True | False)
        "False"
>
<!-- ***** -->
<!-- Resource type element -->
<!-- ***** -->
<ELEMENT Resource type
    EMPTY
>
<ATTLIST Resource type
    Name
        #REQUIRED
>
<!-- ***** -->
<!-- WizardPage element -->
<!-- ***** -->
<ELEMENT WizardPage
    (
        TextLine |
        TextArea |
        SelectList:em
    )+
>
<ATTLIST WizardPage
    SElementCommon:
    WizardPageName
        #REQUIRED
    PolicyTemplateVariableForSkipRule
        #IMPLIED
    SkipRule
        (GT | GE | LT | LE | EQ | NE)
        "EQ"
    CompareAs
        (String | Integer)
        "Integer"

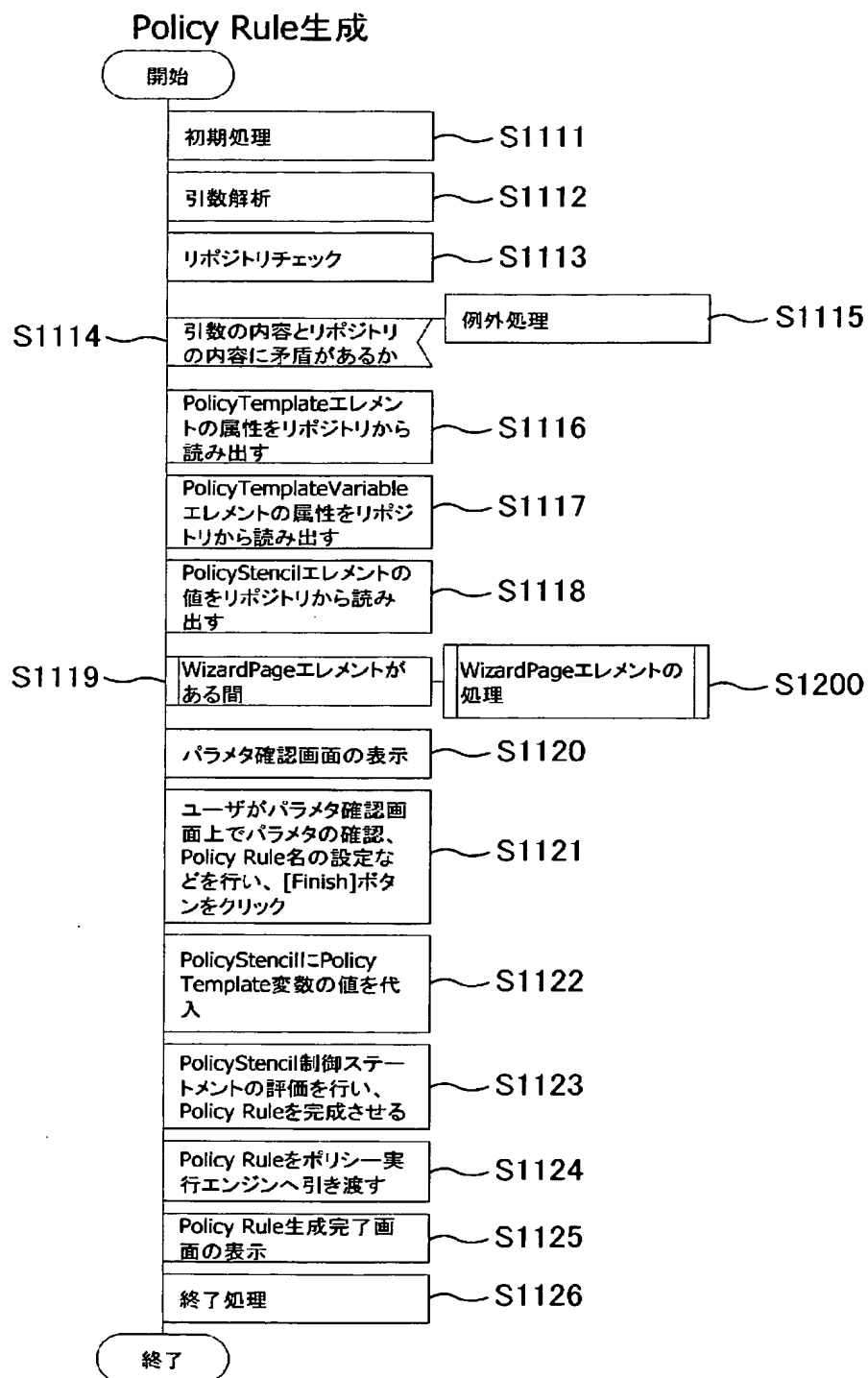
```



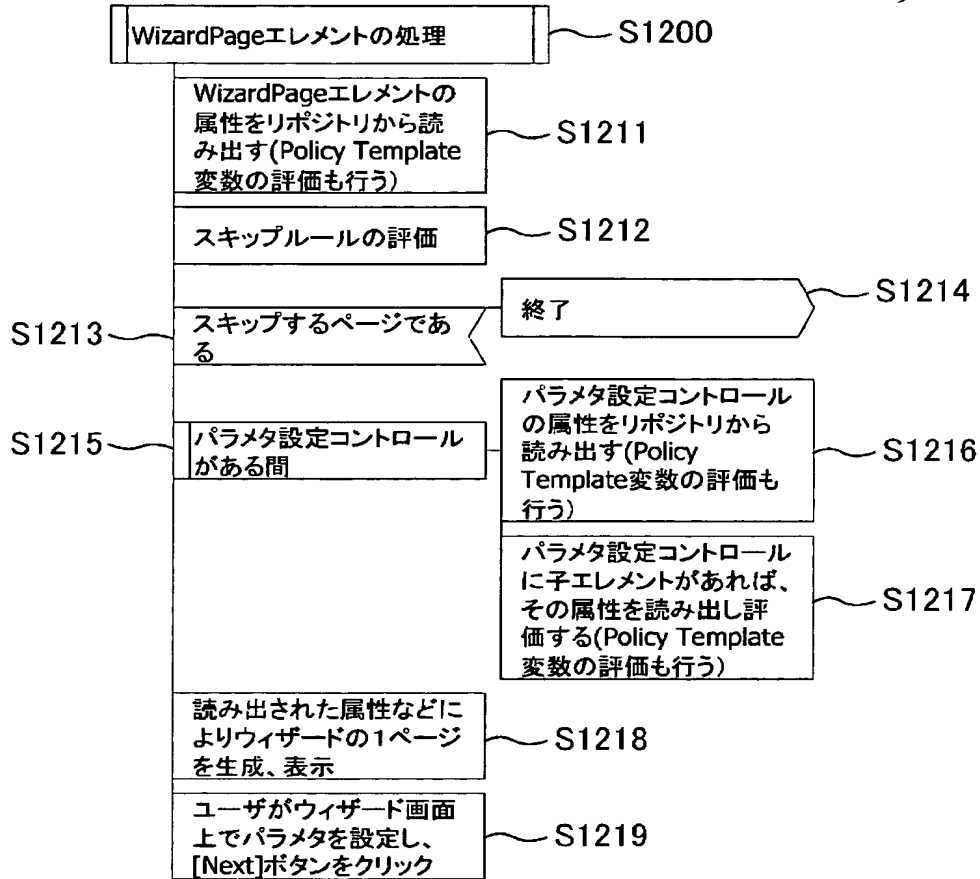
【図 10】



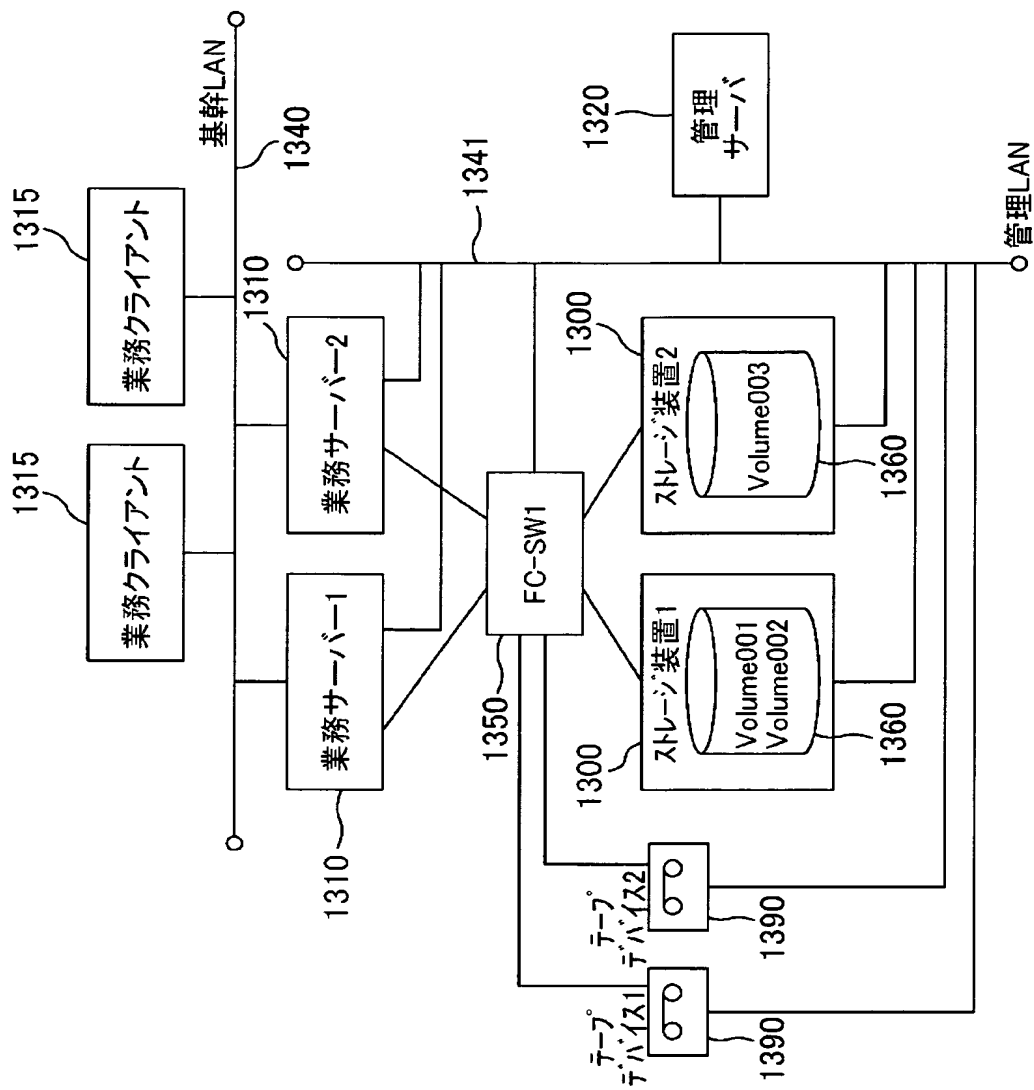
【図 11】



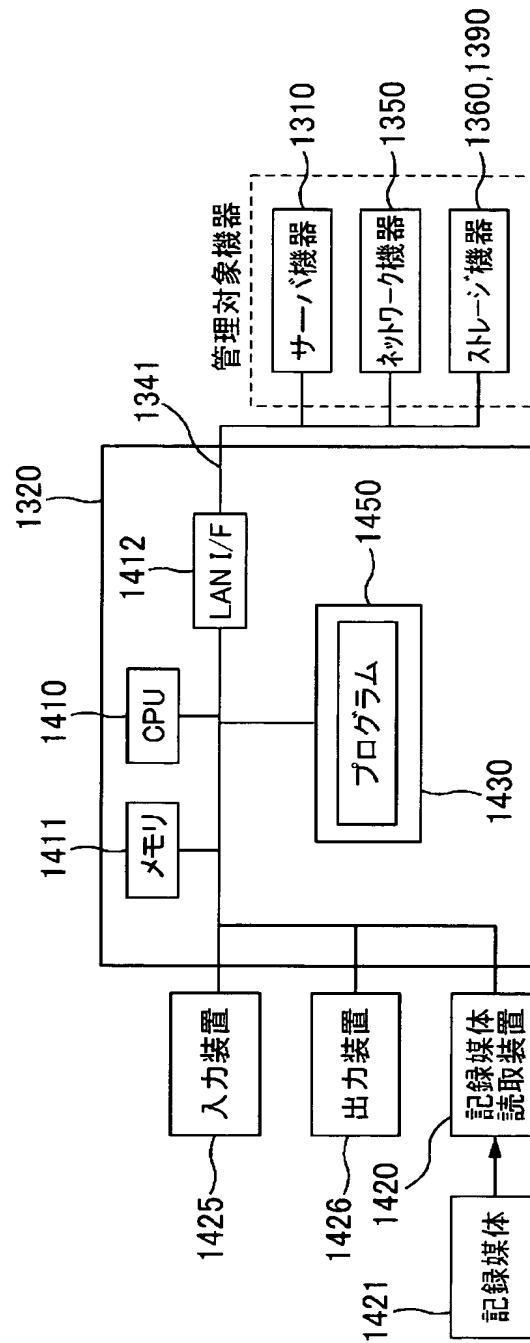
【図 12】



【図 13】



【図 14】



【書類名】 要約書

【要約】

【解決手段】 情報処理装置を含んで構成される情報処理システムにおけるジョブ管理方法において、ジョブ定義文の雛形と、ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とを記憶するステップと、前記ジョブ定義文の雛形と、前記ジョブ定義文を設定させるためのユーザインタフェースを規定するデータの雛形とに基づいて、前記ユーザインタフェースによりユーザが設定する内容に基づいてジョブ定義文を生成する処理を実行させるためのデータを生成するステップと、生成された前記データに基づいて前記処理を実行することによりジョブ定義文を生成するステップと、を備えるジョブ管理方法を提供する。

【選択図】 図 1

特願 2 0 0 3 - 0 5 4 2 7 7

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 5 1 0 8 ]

1 . 変更年月日

1 9 9 0 年 8 月 3 1 日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台 4 丁目 6 番地

氏 名

株式会社日立製作所